What is claimed:

1. A method for creating a neural network based on a plurality of training cases, for the detection of medical events from a record of instrument feature values, comprising:

collecting the plurality of training cases, wherein each training case has an input state and a corresponding output value;

constructing the neural network based on the training cases; and

training the neural network.

2. The method of claim 1, wherein the step of collecting the plurality of training cases further comprises:

selecting a plurality of time epochs from a record of instrument feature values; and

indicating an output value for each selected time epoch.

3. The method of claim 2, the step of collecting the plurality of training cases further comprises:

selecting a configuration of instrument features; and wherein the step of constructing the neural network based on the training cases comprises:

defining the neural network topology based on the input values and output values of the plurality of training cases; and

determining a kernel width value.

4. The method of claim 3, wherein the step of training the neural network includes determining an optimal kernel width value by minimizing prediction error of the neural network.

5. The method of claim 4, wherein the step of training the neural network further comprises:

determining an optimal input feature kernel width value for each input feature based on the determined optimal kernel width value.

6. The method of claim 3, wherein the neural network is a probabilistic neural network.

7. The method of claim 3, wherein the neural network is a generalized regression neural network.

8. The method of claim 3, wherein determining the kernel width value is based on a population statistic of the plurality of training cases.

9. The method of claim 8, wherein determining the kernel width value is based at least in part on the mathematical term of the number of training cases raised to an exponent power of about negative one-fifth.

10. The method of claim 9, wherein determining the kernel width value is based on the population distribution of the plurality of training cases.

11. The method of claim 10, wherein the population distribution of the plurality of training cases is approximately Normal.

12. The method of claim 3, further comprising normalizing the input values of the plurality of training cases based on the standard deviation for each input feature, and wherein the step of determining the kernel width value comprises defining the kernel width value to be a number in the range 0.1 to 1.0.

13. The method of claim 3, wherein collection of the plurality of training cases further comprises:

normalizing the input values of the plurality of training cases based on the standard deviation for each input feature.

14. A method for compressing a neural network, comprising:

determining a plurality of partitions based on the pattern layer nodes of the neural network wherein each partition comprises a plurality of groups of pattern layer nodes;

selecting one of the plurality of partitions based on a partition metric; and

for each group of pattern layer nodes within the selected partition:

replacing the group of pattern layer nodes with a compressed pattern layer node; and

adjusting the link weights between the compressed pattern layer node and any summation layer nodes to reflect the number of replaced pattern layer nodes.

15. The method of claim 14, wherein the partition metric comprises determining a BIC value for each partition.

16. The method of claim 15, wherein the partition metric comprises selecting the maximum BIC value.

17. The method of claim 14, wherein the partition metric comprises determining an error value for each partition.

18. The method of claim 14, wherein the partition metric comprises determining a compression ratio for each partition.

19. The method of claim 14, wherein the partition metric comprises determining a Minimum Description Length for each partition.

20. The method of claim 14, wherein the partition metric comprises determining a BIC value, an error value, and a compression ratio value for each partition.

21. The method of claim 20, wherein the K-means clustering method is applied to determine a plurality of partitions.

22. The method of claim 20, wherein the hierarchical clustering method is used to determine the plurality of partitions.

23. The method of claim 22, wherein the step of determining a plurality of partitions comprises applying the hierarchical clustering method to create partitions containing between about 1 and about 20 groups.

24. The method of claim 14, wherein selecting one of the determined plurality of partitions based on a partition metric comprises:

determining, for each partition within the determined plurality of partitions, a centroid value for each group of pattern layer nodes within that partition.

25. The method of claim 24, wherein selecting one of the determined plurality of partitions based on a partition metric further comprises:

determining, for each partition within the determined plurality of partitions, a covariance value for each group of pattern layer nodes within that partition.

26. A method for combining a first constituent neural network with a second constituent original neural network, comprising:

determining which pattern layer nodes of the constituent neural networks are redundant;

creating a combined neural network by adding non-redundant pattern layer nodes of one constituent neural network to the pattern layer of the other constituent neural network.

27. The method of claim 26, wherein the second constituent neural network has extra input nodes not present in the first constituent neural network, and further comprising:

adding input feature values to the training cases of the first constituent neural network, wherein the added input feature values correspond to the extra input nodes, based on the raw instrument values corresponding to the training cases of the first constituent neural network.

28. A method for incrementally updating a neural network based on a first training case, comprising:

reconfiguring the neural network based on a first training case without retraining the neural network; and

applying the neural network to detect an event in a record of values.

29. The method of claim 28, wherein reconfiguring the neural network comprises adding a first pattern layer node to the neural network based on the first training case.

30. The method of claim 29, further comprising creating the first training case based on the record of values.

31. A method for incrementally updating a neural network based on correcting a prediction error, comprising:

applying the neural network to generate a first output value based on a first input state;

detecting a first prediction error in the first output value;

creating a first training case based on the first input state wherein the first training case corrects the first prediction error;

reconfiguring the neural network based on the first training case without retraining the neural network; and

applying the neural network to generate a second output value based on a second input state.

32. The method of claim 31, wherein reconfiguring the detection module further comprises adding a first pattern layer node to the neural network based on the first training case.

33. A method for updating a detection module configured to classify input states into event classes wherein the detection module is initially incapable of correctly classifying a first input state, comprising:

creating a training case, by selecting a second input state and associating it with an event class; and

reconfiguring the detection module in real-time to correctly classify the first input state based on the training case.

34. A method for updating a medical event detection module configured to classify input states into classes of medical states wherein the detection module is initially incapable of correctly classifying a first input state, comprising:

creating a training case, by selecting a second input state and associating it with a medical state; and

reconfiguring the detection module in real-time to correctly classify the first input state based on the training case.

35. A method for updating a medical event detection module configured to classify input states into classes of medical states wherein the detection module initially incorrectly detects a first medical state based on a first input state, comprising:

creating a training case, by selecting a second input state and associating it with a second medical state; and

reconfiguring the detection module in real-time to detect that the first input state does not correspond to the first medical state.

36. A method for updating a medical event detection module configured to classify input states into classes of medical states wherein the detection module initially fails to detect a first medical event from a first input state, comprising:

creating a training case, by selecting a second input state and associating it with a second medical event; and

reconfiguring the detection module in real-time to correctly detect the first medical event from the first input state.

37. A method for updating a detection module in real-time based on correcting a classification error, comprising:

applying the detection module to classify a first input state into a first event class;

detecting that the detection module incorrectly classified the first input state into the first event class;

creating a first training case by associating the first input state with a second event class; and

reconfiguring the detection module in real-time based on the first training case.

38. A method for updating a neural network, comprising:

transmitting a training case to a user; and

reconfiguring the neural network based on the training case.

39. The method of claim 38, wherein the user receives the training case from a server.

40. The method of claim 38, wherein the user receives the training case from a remote user.

41. The method of claim 38, wherein the training case is transmitted over a network.

42. The method of claim 41, wherein the network is the Internet.

43. The method of claim 41, wherein the network is a LAN.

44. The method of claim 42, wherein the network is a wireless network.

45. The method of claim 38, wherein the training case is transmitted when the second user downloads the training case from a central network server.

46. The method of claim 38, wherein the reconfiguration of the neural network is accomplished in real time.

47. A method for updating a second neural network, comprising:

creating a training case;

transmitting the training case to a second user; and

reconfiguring a second neural network based on the training case.

48. The method of claim 47, wherein the training case is created by a first user on a first neural network.

49. The method of claim 47, wherein the first user is remotely located.

50. The method of claim 47, wherein the training case is transmitted over a network.

51. The method of claim 50, wherein the network is the Internet.

52. The method of claim 50, wherein the network is a LAN.

53. The method of claim 51, wherein the network is a wireless network.

54. The method of claim 47, wherein the training case is transmitted when the second user downloads the training case from a central network server.

55. The method of claim 47, wherein the reconfiguration of the second neural network using the training case is accomplished in real time.

56. A method for updating a first neural network and a second neural network, comprising:

creating a training case using the first neural network;

reconfiguring a first neural network in real-time based on the training case;

transmitting the training case created by the first neural network to a second user; and

reconfiguring a second neural network based on the training case.

57. The method of claim 54, wherein the first user is remotely located.

58. The method of claim 56, wherein the training case is transmitted over a network.

59. The method of claim 58, wherein the network is the Internet.

60. The method of claim 58, wherein the network is a LAN.

61. The method of claim 58, wherein the network is a wireless network.

62. The method of claim 58, wherein the training case created using the first neural network is stored on a central network server.

63. The method of claim 62, wherein the training case is transmitted when the second user downloads the training case from a central network server.

64. The method of claim 58, wherein the reconfiguration of the second neural network using the training case is accomplished in real time.

65. A method for updating a first neural network and a second neural network, comprising:

creating a training case;

reconfiguring a first neural network in real-time based on the training case;

transmitting the training case to a second user; and

reconfiguring a second neural network in real-time based on the training case.

66. The method of claim 65, wherein the neural network utilized by the second user is remotely located from the neural network used to create the training case.

67. The method of claim 65, wherein the training case is transmitted over a network.

68. The method of claim 67, wherein the network is the Internet.

69. The method of claim 67, wherein the network is a LAN.

70. The method of claim 67, wherein the network is a wireless network.

71. The method of claim 65, wherein the training case created using the first neural network is stored on a central network server.

72. The method of claim 71, wherein the training case is transmitted when the second user downloads the training case from a central network server.

73. A method for updating a first neural network and a second neural network, comprising:

creating a training case;

reconfiguring a first neural network in real-time based on the training case; and

transmitting the training case to a receiving module, wherein the receiving module is configured to reconfigure a second neural network in real-time based on the training case.

74. A method for updating a first neural network and a second neural network, comprising:

receiving a training case from a transmitting module, wherein the transmitting module is configured to reconfigure a first neural network in real-time based on the training case; and

reconfiguring the second neural network in real-time based on the training case.

75. A method for updating a first neural network and a second neural network, comprising:

receiving a training case from a transmitting module, wherein the transmitting module is configured to reconfigure a first neural network in real-time based on the training case; and

transmitting the training case to a receiving module, wherein the receiving module is configured to reconfigure a second neural network in real-time based on the training case.

76. A method for updating a second neural network, comprising:

creating a second training case;

reconfiguring the second neural network in real-time based on the second training case;

receiving a first training case; and

further reconfiguring the second neural network in real-time based on the first training case.

77. A system for updating a neural network, comprising:

a network interface configured to interface the system with a network;

a distribution authority coupled to the network interface, the distribution authority configured to receive an update from a first detection module via the network interface, and store the update;

wherein the update comprises a training case.

78. The system of claim 77, wherein the distribution authority is further configured to transmit the update to a second detection module via the network interface.

79. The system of claim 78, wherein the second detection module is configured to reconfigure a second neural network in real-time based on the update.

80. The system of claim 77, wherein the update further comprises a plurality of raw instrument values corresponding to the training case.

81. The system of claim 77, wherein the distribution authority is further configured to determine whether to store and/or transmit the update.